Dealing with Execution Uncertainty in the Continuous Double Auction

Gert van Valkenhoef

University of Groningen

2009-04-15

university of groningen

Contents

What is Resource Allocation?

Resource Allocation

To assign available resources in an economic way. [Wikipedia]

What is Resource Allocation?

Resource Allocation

To assign available resources in an economic way. [Wikipedia]

Computational Resource Allocation

What is Resource Allocation?

Resource Allocation

To assign available resources in an economic way. [Wikipedia]

Computational Resource Allocation

- CPU time slice allocation

What is Resource Allocation?

Resource Allocation

To assign available resources in an economic way. [Wikipedia]

Computational Resource Allocation

- CPU time slice allocation
- Wireless channel allocation

What is Resource Allocation?

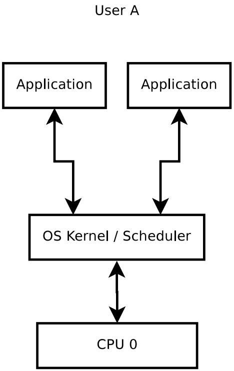
Resource Allocation

To assign available resources in an economic way. [Wikipedia]

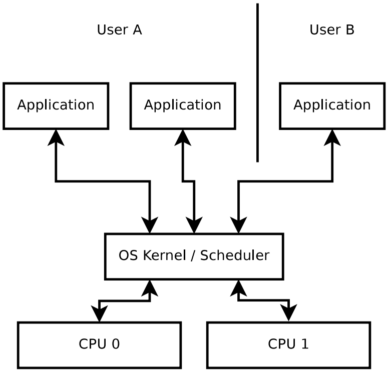
Computational Resource Allocation

- CPU time slice allocation
- Wireless channel allocation
- Network bandwidth allocation

CPU time slice allocation



CPU time slice allocation



CPU time slice allocation

This case is easy:

- There is a clear central authority (the kernel)
- The users are usually the owners of the system

Why Grid Computing?

- Processing enormous quantities of data
- Collaboratively tackling difficult challenges
- In general: computational resource sharing

Why Grid Computing?

- Processing enormous quantities of data
- Collaboratively tackling difficult challenges
- In general: computational resource sharing

LHC Computing Grid (LCG)

- Processes 300 MB/s filtered data (from 300 GB/s raw)
- Combines 140 computing centers in 33 countries

What is Grid Computing

What is Grid Computing

- Organises geographically separated computing resources

What is Grid Computing

- Organises geographically separated computing resources
- Different computing centers are combined by high-speed links

What is Grid Computing

- Organises geographically separated computing resources
- Different computing centers are combined by high-speed links
- To form Virtual Organisations (VOs)

What is Grid Computing

- Organises geographically separated computing resources
- Different computing centers are combined by high-speed links
- To form Virtual Organisations (VOs)
- VO members use resources according to predefined rules

What is Grid Computing

- Organises geographically separated computing resources
- Different computing centers are combined by high-speed links
- To form Virtual Organisations (VOs)
- VO members use resources according to predefined rules
- Computing power available on-demand: utility computing

What is Grid Computing

- Organises geographically separated computing resources
- Different computing centers are combined by high-speed links
- To form Virtual Organisations (VOs)
- VO members use resources according to predefined rules
- Computing power available on-demand: utility computing
- Analogy: the electricity grid

Grid Resource Allocation

Grid resource allocation is harder:

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority
- Much larger problem space

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority
- Much larger problem space

Possible solutions:

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority
- Much larger problem space

Possible solutions:

- Create central resource broker

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority
- Much larger problem space

Possible solutions:

- Create central resource broker
  - Risk: bottleneck

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority
- Much larger problem space

Possible solutions:

- Create central resource broker
  - Risk: bottleneck
  - Risk: single point of failure

Grid Resource Allocation

Grid resource allocation is harder:

- No clear central authority
- Much larger problem space

Possible solutions:

- Create central resource broker
  - Risk: bottleneck
  - Risk: single point of failure
- Decentralised resource allocation

Grid resource allocation is ‘easy’:

Grid resource allocation is ‘easy’:

- VOs created in cooperative setting

Grid resource allocation is 'easy':

- VOs created in cooperative setting
- Hence, conflict of interest not a problem

Volunteer Computing

- Volunteers sign up, install software on their computer
- Runs when computer is idle
- Fetches and completes work packages
- Volunteer earns reputation ‘points’

Why Volunteer Computing?

IBM Roadrunner

- #1 super computer
- 1.1 PetaFLOPS (130.000 cores)

Why Volunteer Computing?

IBM Roadrunner

- #1 super computer
- 1.1 PetaFLOPS (130.000 cores)

BOINC (Berkeley)

- Runs many projects, including SETI@Home
- 1.5 PetaFLOPS

Why Volunteer Computing?

IBM Roadrunner

- #1 super computer
- 1.1 PetaFLOPS (130.000 cores)

BOINC (Berkeley)

- Runs many projects, including SETI@Home
- 1.5 PetaFLOPS

Folding@Home (Stanford)

- Protein folding research (medical)
- Utilises CPU, GPU and PS3
- 5.0 PetaFLOPS (425.000 clients)

Resource Allocation

- Signed up to specific project
- So: most important tasks for project assigned first

Resource Allocation

- Signed up to specific project
- So: most important tasks for project assigned first
- However: output from nodes unreliable

Resource Allocation

- Signed up to specific project
- So: most important tasks for project assigned first
- However: output from nodes unreliable
- Results must be verified

Resource Allocation

- Signed up to specific project
- So: most important tasks for project assigned first
- However: output from nodes unreliable
- Results must be verified
- Application to time-critical problems difficult

Large-scale, open, distributed systems

Large-scale, open, distributed systems

- Combine resources in ad-hoc fashion
- May include:

Large-scale, open, distributed systems

- Combine resources in ad-hoc fashion
- May include:
    - University computing centres

Large-scale, open, distributed systems

- Combine resources in ad-hoc fashion
- May include:
  - University computing centres
  - Competing corporations

Large-scale, open, distributed systems

- Combine resources in ad-hoc fashion
- May include:
  - University computing centres
  - Competing corporations
  - Consumer PCs

Large-scale, open, distributed systems

- Combine resources in ad-hoc fashion
- May include:
  - University computing centres
  - Competing corporations
  - Consumer PCs
  - Network of workstations that are idle overnight

Problems

Problems

- Large problem size

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Solutions

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Solutions

- Decentralised resource allocation

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Solutions

- Decentralised resource allocation
- Market-based mechanisms

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Solutions

- Decentralised resource allocation
- Market-based mechanisms
- (Some) market-based mechanisms

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Solutions

- Decentralised resource allocation
- Market-based mechanisms
- (Some) market-based mechanisms
- Continuous market-based mechanism

Problems

- Large problem size
- Non-cooperative participants – how to prioritise jobs?
- Multiple users, multiple providers
- Users, providers enter/leave system at any time
- Unreliable participants

Solutions

- Decentralised resource allocation
- Market-based mechanisms
- (Some) market-based mechanisms
- Continuous market-based mechanism
- Trust-based mechanisms

Solutions

- Decentralised resource allocation
- Market-based mechanisms
- (Some) market-based mechanisms
- Continuous resource allocation mechanisms
- Trust-based mechanisms

Solutions

- Decentralised resource allocation
- Market-based mechanisms
- (Some) market-based mechanisms
- Continuous resource allocation mechanisms
- Trust-based mechanisms

Continuous Double Auction

Decentralised, continuous, market-based resource allocation

Solutions

- Decentralised resource allocation
- Market-based mechanisms
- (Some) market-based mechanisms
- Continuous resource allocation mechanisms
- Trust-based mechanisms

Continuous Double Auction

Decentralised, continuous, market-based resource allocation

- Has the first four properties required
- Lacks trust-based decision making

Contents

The Continuous Double Auction: Setting

- Market for trading a specific commodity
- Multiple buyers and sellers
- Buyers and sellers have their own valuations (limit prices)
- Traders shout prices (offers)
  - Bid: offer to buy
  - Ask: offer to sell

The Continuous Double Auction: Mechanism

The Continuous Double Auction: Mechanism

- Order Books

The Continuous Double Auction: Mechanism

- Order Books
- Shout Accepting Rule

The Continuous Double Auction: Mechanism

- Order Books
- Shout Accepting Rule
- Clearing Rule

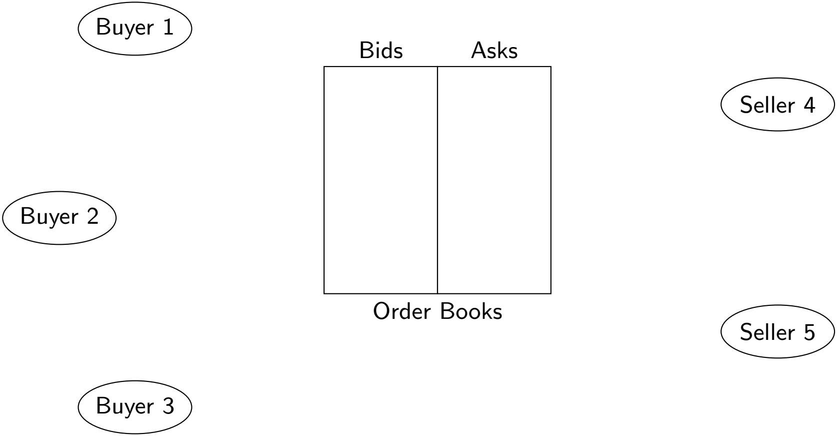
The Continuous Double Auction: Mechanism

- Order Books
- Shout Accepting Rule
- Clearing Rule
- Pricing Rule

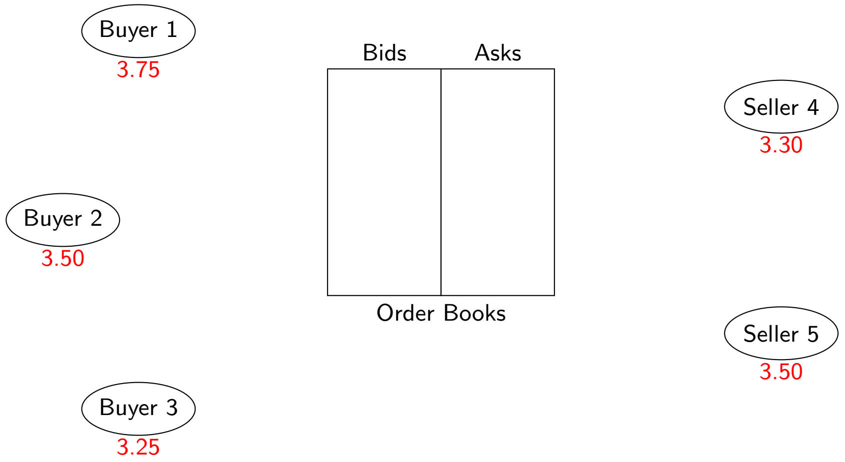
The Continuous Double Auction: Mechanism

- Order Books
- Shout Accepting Rule
- Clearing Rule
- Pricing Rule
- Information Revelation Rule

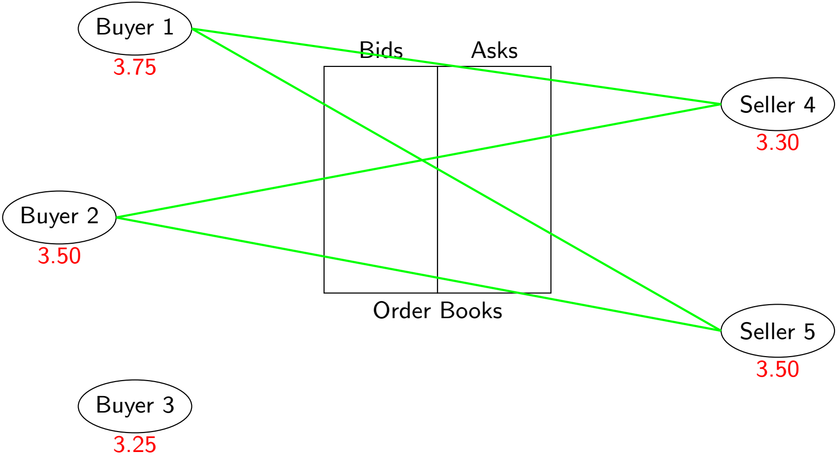
Example


Buyer 1
Bids
Asks
Seller 4
Buyer 2
Order Books
Seller 5
Buyer 3

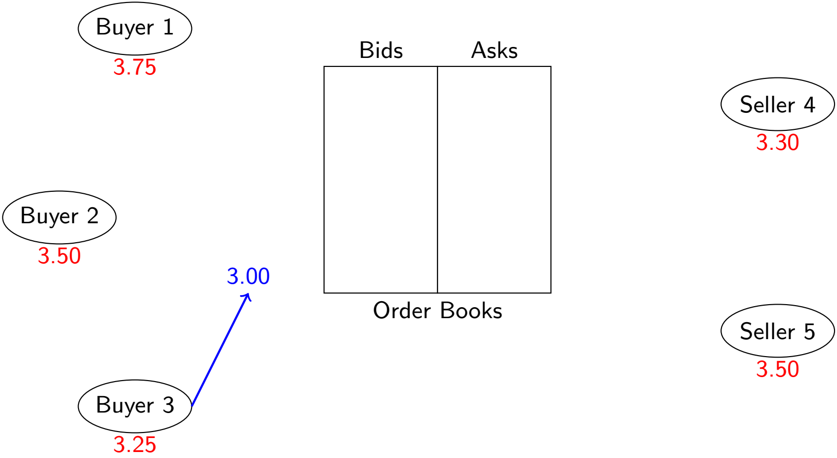
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
Order Books
Seller 4
3.30
Seller 5
3.50

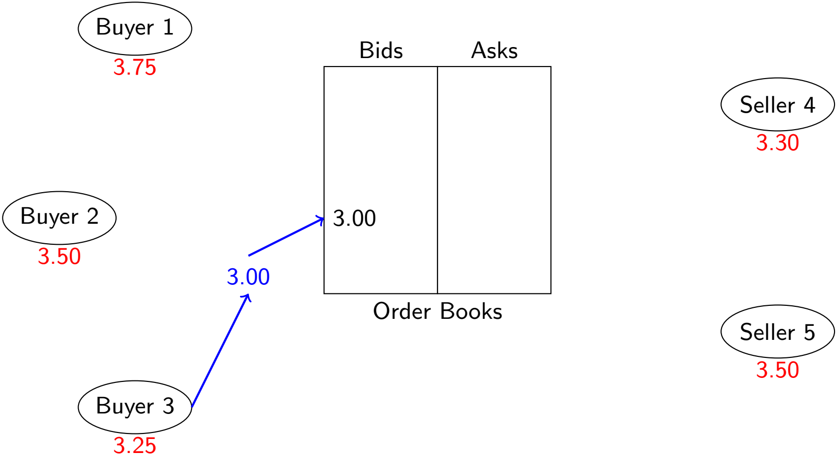
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
Order Books
Seller 4
3.30
Seller 5
3.50

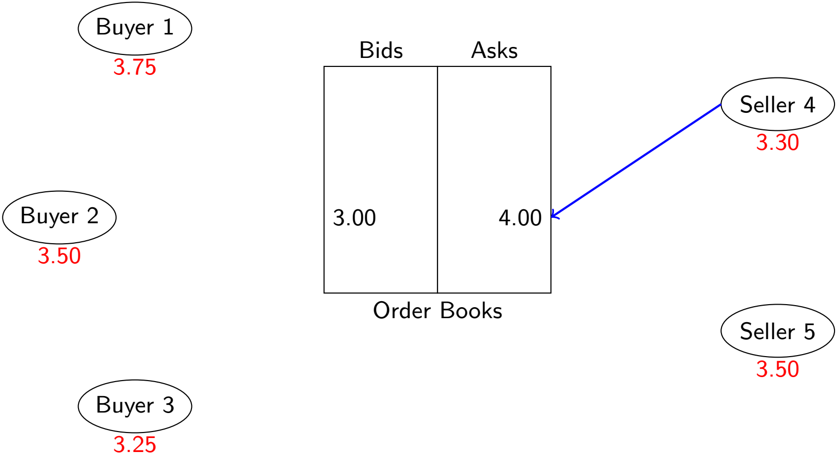
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
3.00
Bids
Asks
Order Books
Seller 4
3.30
Seller 5
3.50

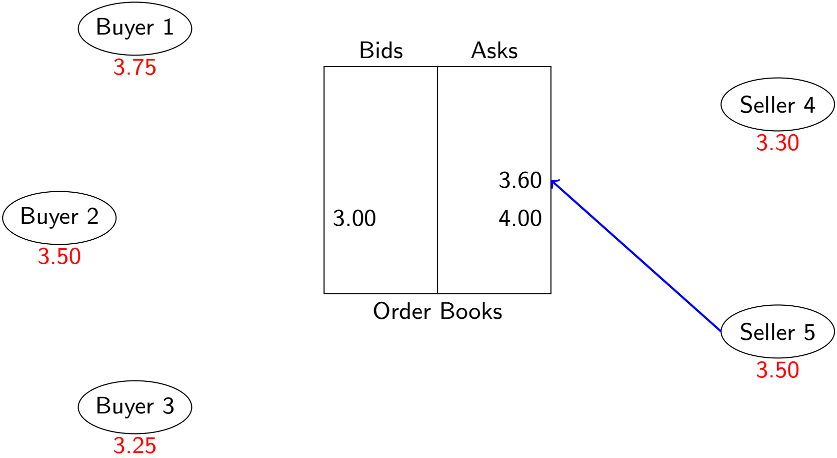
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
3.00
Bids
Asks
3.00
Order Books
Seller 4
3.30
Seller 5
3.50

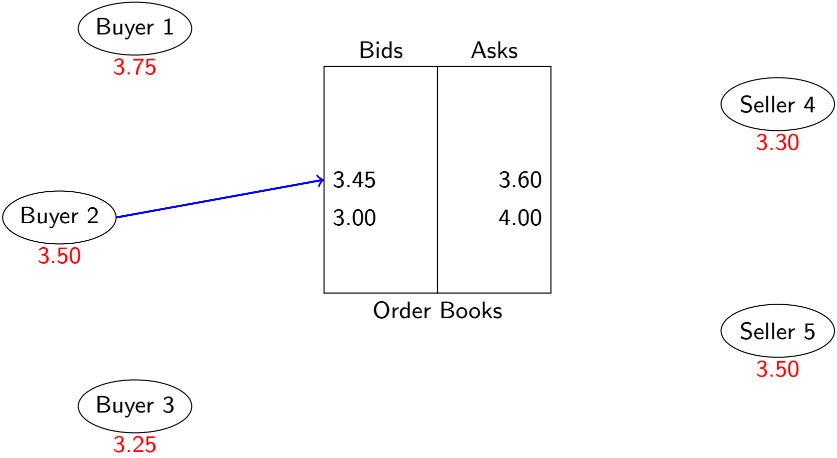
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.00
4.00
Order Books
Seller 4
3.30
Seller 5
3.50

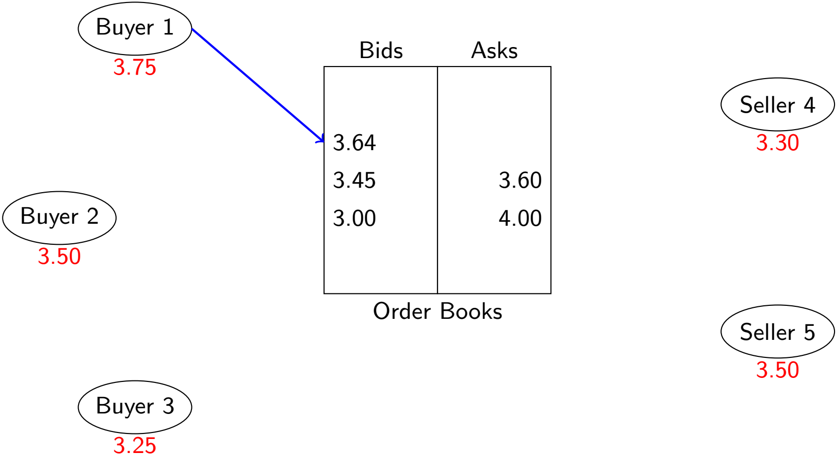
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.60
3.00
4.00
Order Books
Seller 4
3.30
Seller 5
3.50

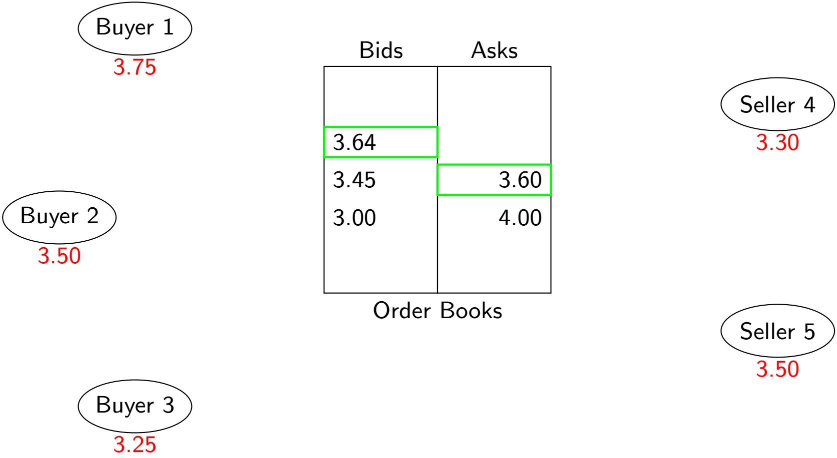
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.45
3.00
3.60
4.00
Order Books
Seller 4
3.30
Seller 5
3.50

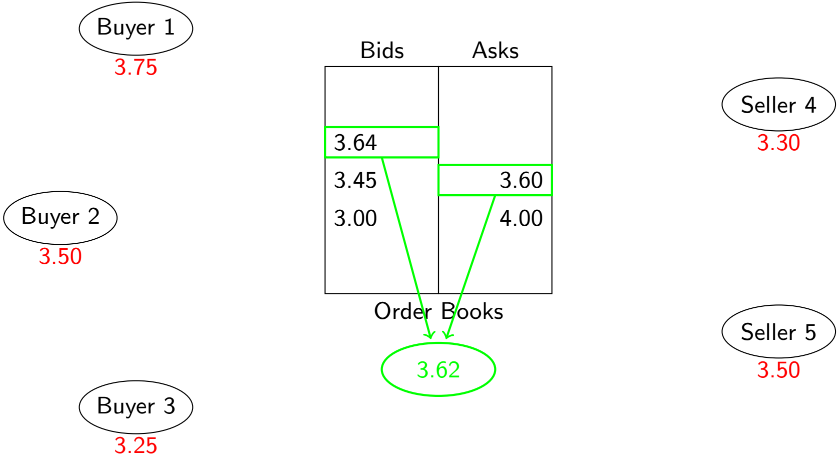
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.64
3.45
3.00
3.60
4.00
Order Books
Seller 4
3.30
Seller 5
3.50

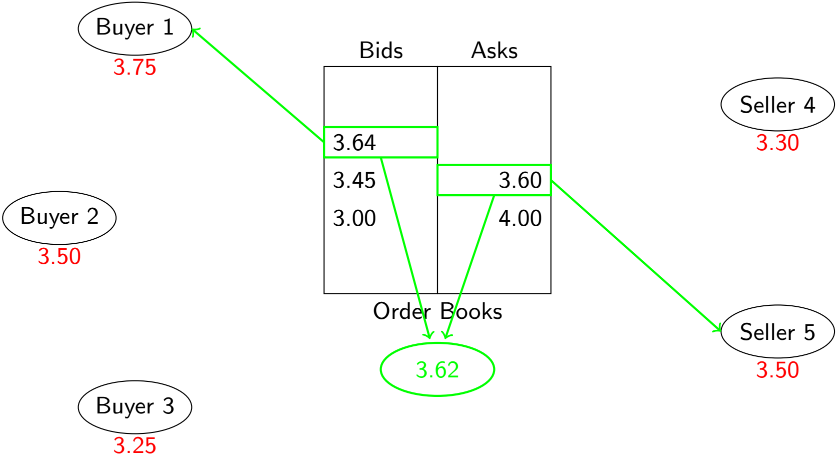
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.64
3.45
3.00
3.60
4.00
Order Books
Seller 4
3.30
Seller 5
3.50

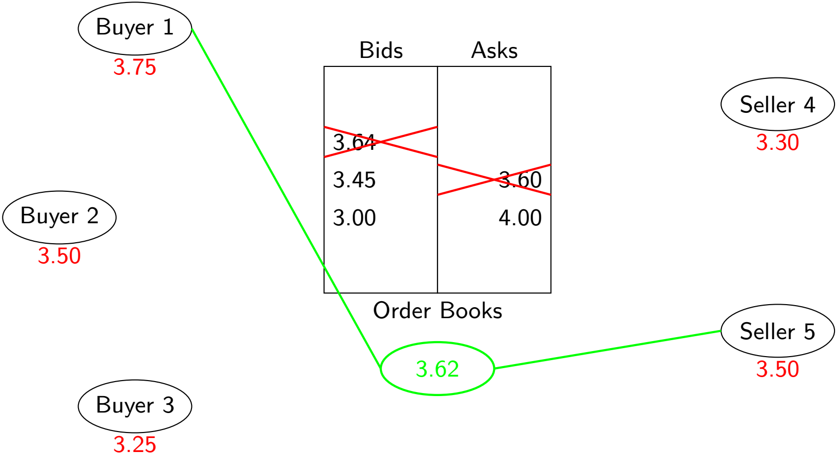
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.64
3.45
3.00
3.60
4.00
Order Books
3.62
Seller 4
3.30
Seller 5
3.50

Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.64
3.45
3.00
3.60
4.00
Order Books
3.62
Seller 4
3.30
Seller 5
3.50

Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
3.64
3.45
3.00
3.60
4.00
Order Books
3.62
Seller 4
3.30
Seller 5
3.50

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75
- Now, assume Seller 5 has only 96% chance of delivering

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75
- Now, assume Seller 5 has only 96% chance of delivering
- Should Buyer 1 still transact?

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75
- Now, assume Seller 5 has only 96% chance of delivering
- Should Buyer 1 still transact?
- Calculate Expected Utility:

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75
- Now, assume Seller 5 has only 96% chance of delivering
- Should Buyer 1 still transact?
- Calculate Expected Utility:

$$\hat{u} = (0.96 \times 3.75 + (1 - 0.96) \times 0) - 3.62 = -0.02$$

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75
- Now, assume Seller 5 has only 96% chance of delivering
- Should Buyer 1 still transact?
- Calculate Expected Utility:

$$\hat{u} = (0.96 \times 3.75 + (1 - 0.96) \times 0) - 3.62 = -0.02$$

- So, Buyer 1 expects to make a loss!

Execution Uncertainty

- Buyer 1 and Seller 5 transact at 3.62
- Buyer 1 has limit price 3.75
- Now, assume Seller 5 has only 96% chance of delivering
- Should Buyer 1 still transact?
- Calculate Expected Utility:

$$\hat{u} = (0.96 \times 3.75 + (1 - 0.96) \times 0) - 3.62 = -0.02$$

- So, Buyer 1 expects to make a loss!
- Hence, the CDA fails when we introduce execution uncertainty

Trust-Based CDA

Commitment in the CDA

When a bid/ask is submitted in the CDA:

- The submitter commits to a transaction at that price
- With *anyone*

Trust-Based CDA

Commitment in the CDA

When a bid/ask is submitted in the CDA:

- The submitter commits to a transaction at that price
- With *anyone*

Commitment in the T-CDA

Trust-Based CDA

Commitment in the CDA

When a bid/ask is submitted in the CDA:

- The submitter commits to a transaction at that price
- With *anyone*

Commitment in the T-CDA

- T-CDA separates commitment from bid/ask

Trust-Based CDA

Commitment in the CDA

When a bid/ask is submitted in the CDA:

- The submitter commits to a transaction at that price
- With *anyone*

Commitment in the T-CDA

- T-CDA separates commitment from bid/ask
- By adding 'commitment phase': extra step

Trust-Based CDA

Commitment in the CDA

When a bid/ask is submitted in the CDA:

- The submitter commits to a transaction at that price
- With *anyone*

Commitment in the T-CDA

- T-CDA separates commitment from bid/ask
- By adding ‘commitment phase’: extra step
- Agents can commit to or reject a match

Trust-Based CDA

Commitment in the CDA

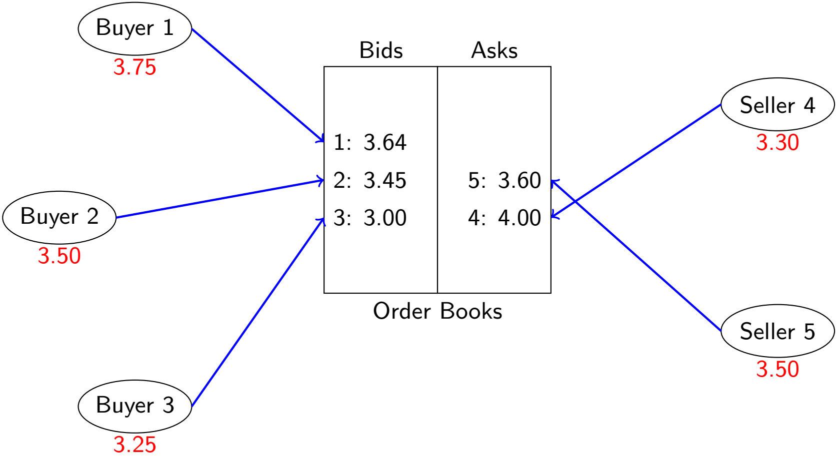
When a bid/ask is submitted in the CDA:

- The submitter commits to a transaction at that price
- With *anyone*

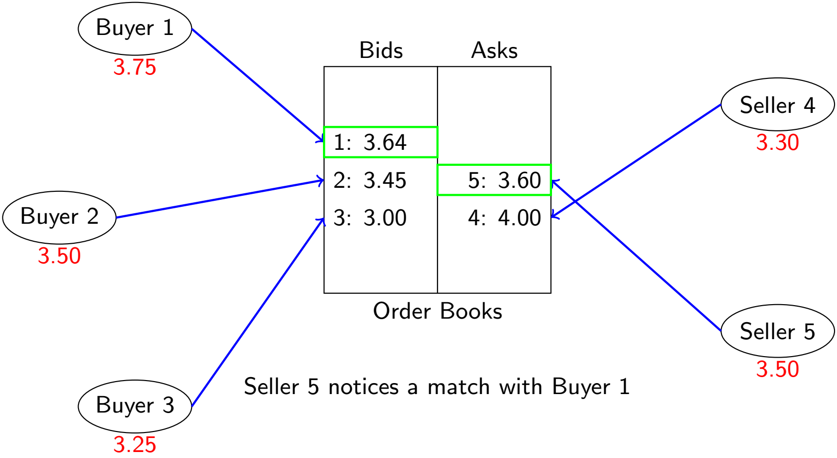
Commitment in the T-CDA

- T-CDA separates commitment from bid/ask
- By adding 'commitment phase': extra step
- Agents can commit to or reject a match
- This allows them to use their *trust* information

Example

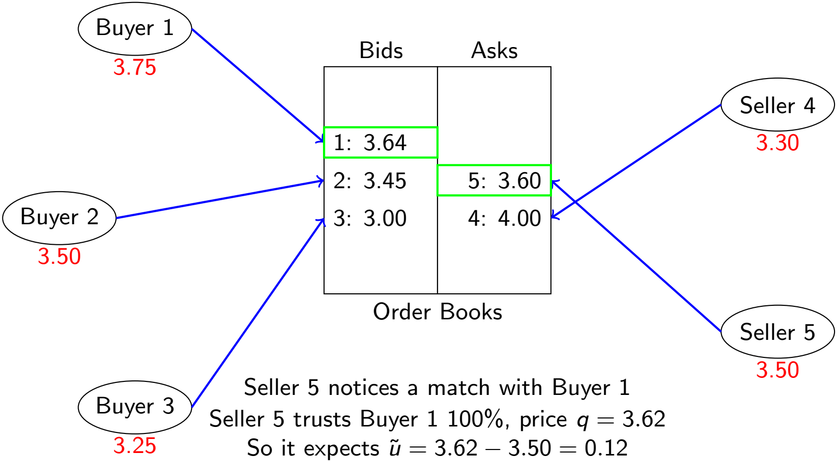

Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.60
4: 4.00
Order Books
Seller 4
3.30
Seller 5
3.50

Example



Seller 5 notices a match with Buyer 1

Example

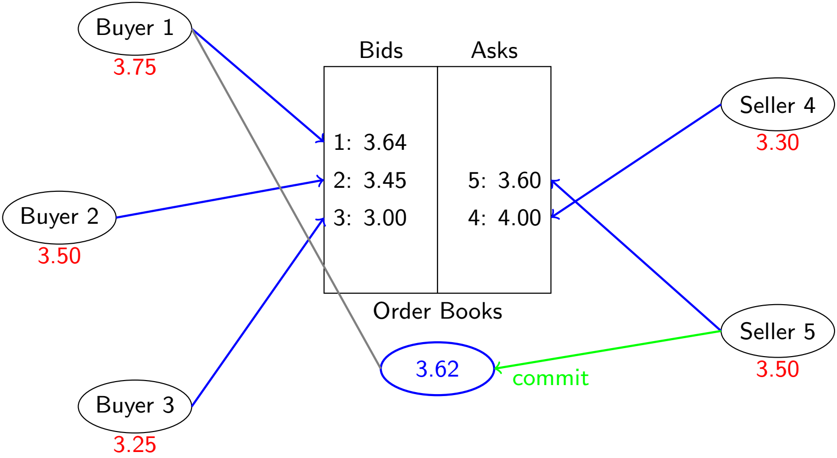


Seller 5 notices a match with Buyer 1

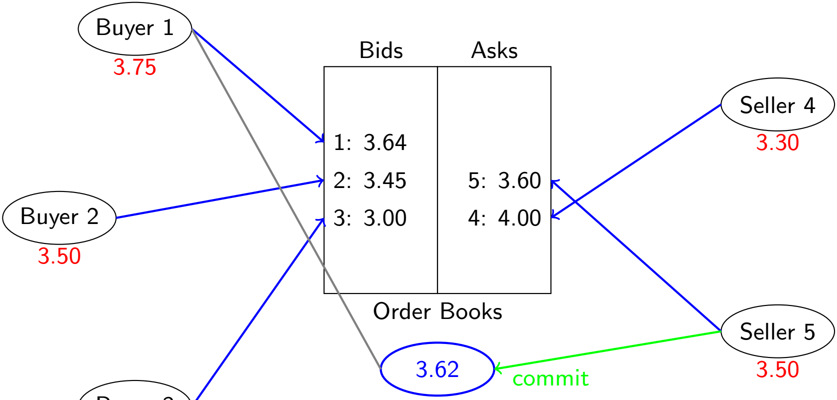
Seller 5 trusts Buyer 1 100%, price $q = 3.62$

So it expects $\tilde{u} = 3.62 - 3.50 = 0.12$

Example

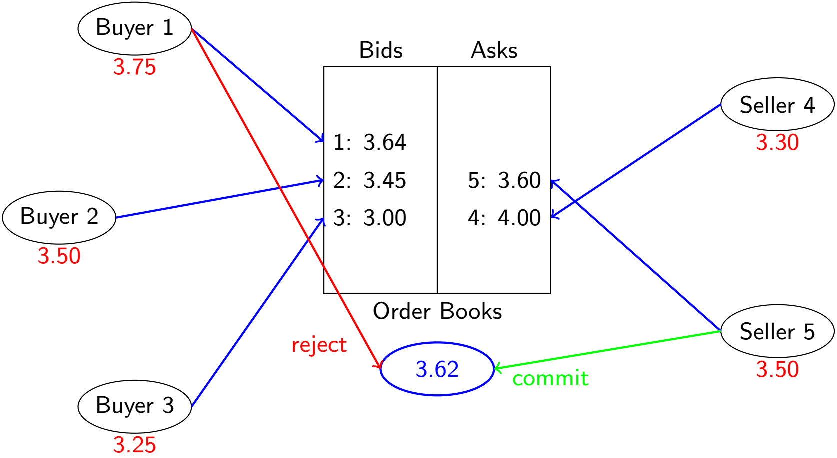

Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.60
4: 4.00
Order Books
Seller 4
3.30
Seller 5
3.50
3.62
commit

Example

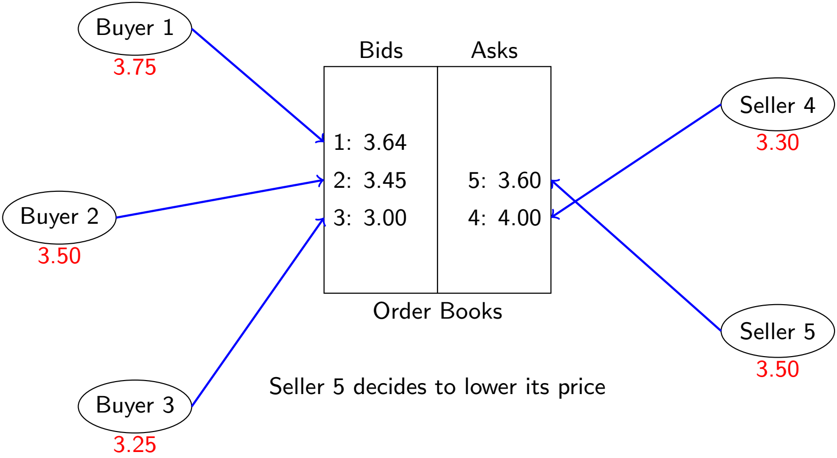


Buyer 5 trusts Seller 5 96%
and expects $\tilde{u} = -0.02$

Example

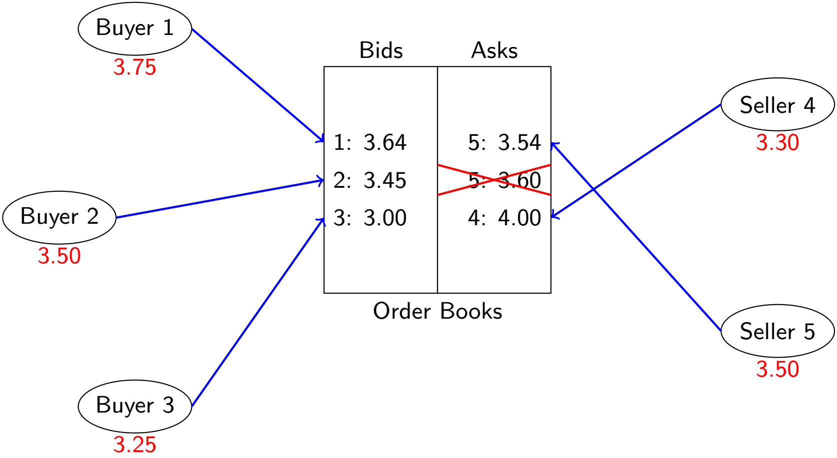

Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.60
4: 4.00
Order Books
Seller 4
3.30
Seller 5
3.50
reject
3.62
commit

Example

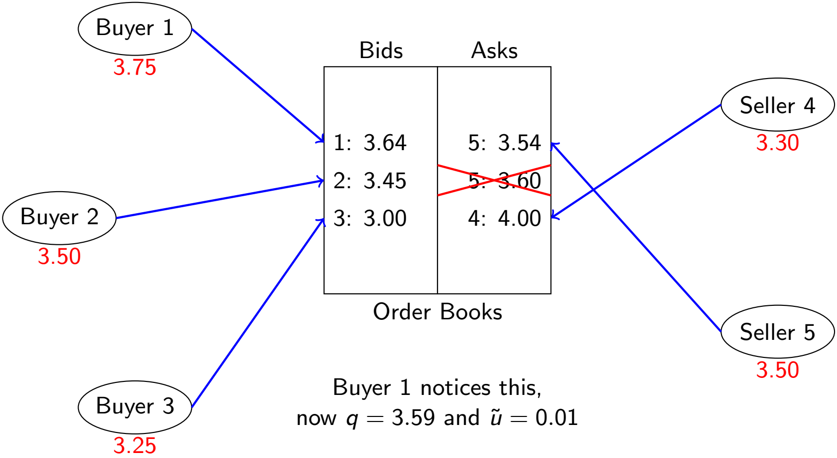


Seller 5 decides to lower its price

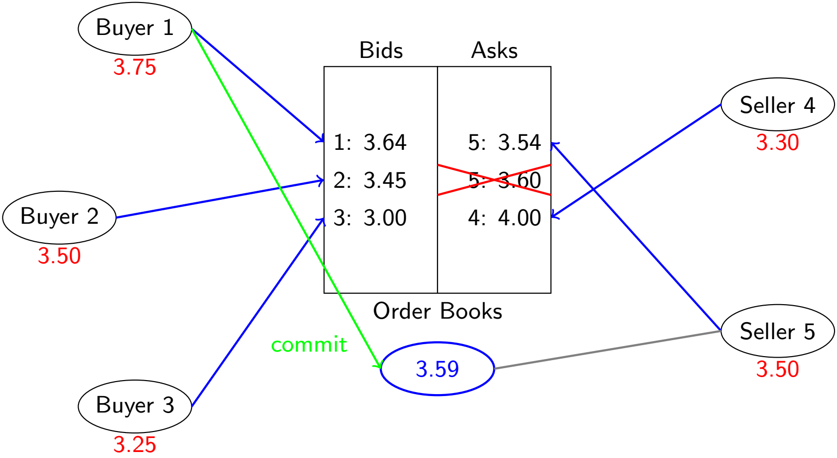
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.54
5: 3.60
4: 4.00
Order Books
Seller 4
3.30
Seller 5
3.50

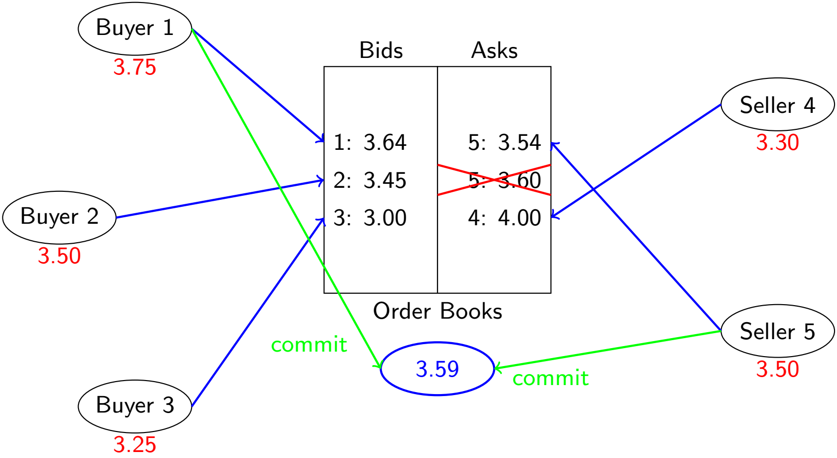
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.54
5: 3.60
4: 4.00
Order Books
Seller 4
3.30
Seller 5
3.50
Buyer 1 notices this,
now $q = 3.59$ and $\tilde{u} = 0.01$

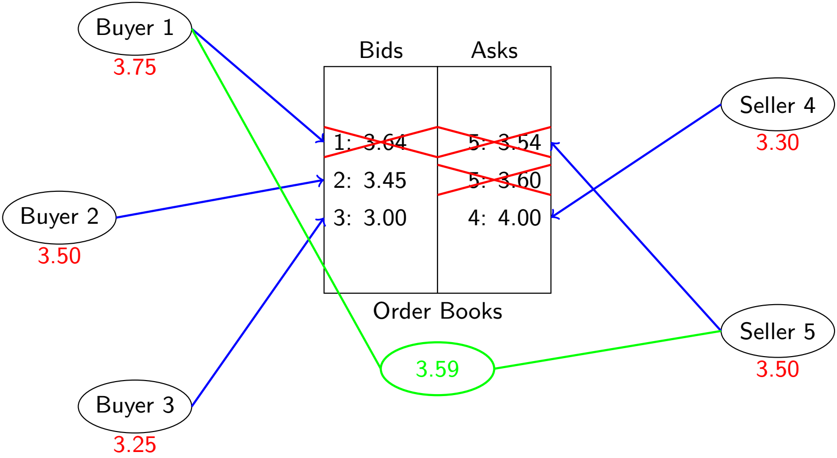
Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.54
5: 3.60
4: 4.00
Order Books
commit
3.59
Seller 4
3.30
Seller 5
3.50

Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.54
5: 3.60
4: 4.00
Order Books
Seller 4
3.30
Seller 5
3.50
commit
3.59
commit

Example


Buyer 1
3.75
Buyer 2
3.50
Buyer 3
3.25
Bids
Asks
1: 3.64
2: 3.45
3: 3.00
5: 3.54
5: 3.60
4: 4.00
Order Books
3.59
Seller 4
3.30
Seller 5
3.50

Contents

Evaluation: Strategies (1 of 2)

Zero Intelligence (ZI)

Evaluation: Strategies (1 of 2)

Zero Intelligence (ZI)

Originally by (Gode and Sunder, 1993). Traders bid randomly:

Evaluation: Strategies (1 of 2)

Zero Intelligence (ZI)

Originally by (Gode and Sunder, 1993). Traders bid randomly:

- Buyers: uniform from $[0, \ell]$

Evaluation: Strategies (1 of 2)

Zero Intelligence (ZI)

Originally by (Gode and Sunder, 1993). Traders bid randomly:

- Buyers: uniform from $[0, \ell]$
- Sellers: uniform from $[\ell, q_{\max}]$

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Based on (Cliff and Bruten, 1997). Adaptive (learning) strategy:

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Based on (Cliff and Bruten, 1997). Adaptive (learning) strategy:

- All market events are information about market conditions

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Based on (Cliff and Bruten, 1997). Adaptive (learning) strategy:

- All market events are information about market conditions
- Based on bids/asks/commits, an agent updates *profit margin*

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Based on (Cliff and Bruten, 1997). Adaptive (learning) strategy:

- All market events are information about market conditions
- Based on bids/asks/commits, an agent updates *profit margin*
- Profit margin and limit price determine bid/ask price

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Based on (Cliff and Bruten, 1997). Adaptive (learning) strategy:

- All market events are information about market conditions
- Based on bids/asks/commits, an agent updates *profit margin*
- Profit margin and limit price determine bid/ask price
- Prices converge to an *equilibrium price*

Evaluation: Strategies (2 of 2)

Trust-Based Zero Intelligence Plus (T-ZIP)

Based on (Cliff and Bruten, 1997). Adaptive (learning) strategy:

- All market events are information about market conditions
- Based on bids/asks/commits, an agent updates *profit margin*
- Profit margin and limit price determine bid/ask price
- Prices converge to an *equilibrium price*
- However, in the T-CDA, this does not always exist

Evaluation: Overview

- Assume all traders are given their trust information (fixed values)
- ZI traders:
    - T-CDA vs CDA
    - T-CDA with imperfect information
    - T-CDA vs optimal centralised solution
- T-ZIP traders:
    - Positive: near 100% efficient when $Var(pos) = 0$
    - Negative: fails when $Var(pos) > 0$

T-CDA vs CDA

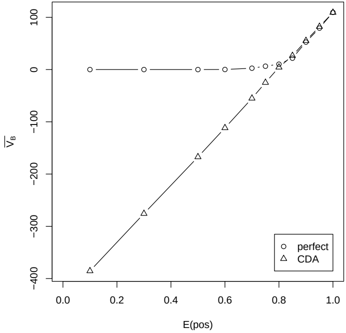


Figure: Utility of the T-CDA (given perfect information) versus CDA, with ZI agents

T-CDA vs CDA

- Traders in the CDA make a loss
- The T-CDA prevents this
- And is always at least as good as the CDA

Imperfect information

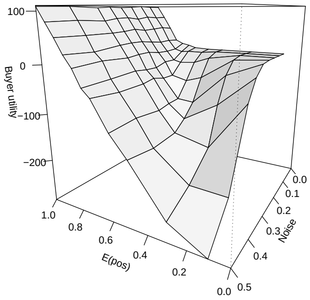


Figure: Performance degrades when $E(pos)$ is lowered and when more noise is added to trust values.

Imperfect information

- Performance degrades linearly when information becomes less accurate

T-CDA vs Optimum

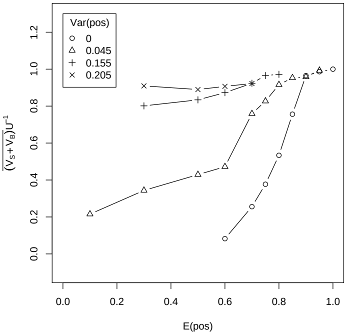


Figure: The normalised utility, or *efficiency* derived by the T-CDA.

T-CDA vs Optimum

- T-CDA with ZI-traders is inefficient for low POS

T-CDA vs Optimum

- T-CDA with ZI-traders is inefficient for low POS
- ZI bidding range too wide when POS is low

T-CDA vs Optimum

- T-CDA with ZI-traders is inefficient for low POS
- ZI bidding range too wide when POS is low
- Hypothesis: this causes inefficiency

T-ZIP: $Var(pos) = 0$

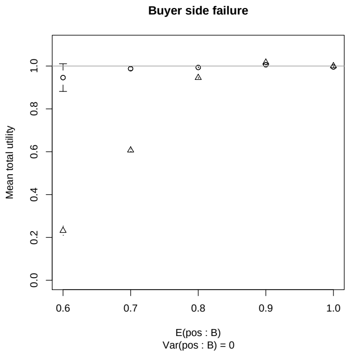


Figure: Buyer side failure, with $Var(pos) = 0$, comparing T-ZIP (circles) and ZI (triangles). Error bars indicate the 95% confidence interval.

T-ZIP: $Var(pos) = 0$

- T-ZIP achieves near-100% efficiency when $Var(pos) = 0$

T-ZIP: $Var(pos) = 0$

- T-ZIP achieves near-100% efficiency when $Var(pos) = 0$
- This confirms that ZI inefficiency is not due to T-CDA

T-ZIP: $Var(pos) = 0$

- T-ZIP achieves near-100% efficiency when $Var(pos) = 0$
- This confirms that ZI inefficiency is not due to T-CDA
- But that ZI bidding is inappropriate

T-ZIP: $Var(pos) > 0$

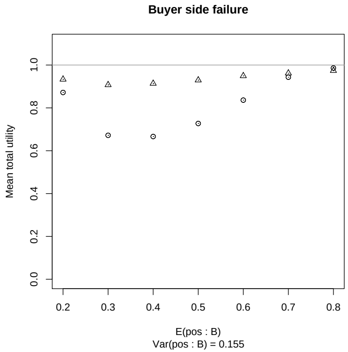


Figure: Buyer side failure, with $Var(pos) = 0.155$, comparing T-ZIP (circles) and ZI (triangles). Error bars indicate the 95% confidence interval.

T-ZIP: $Var(pos) > 0$

- We saw an example where T-ZIP does poorly
- Even ZI does better!

T-ZIP: $Var(pos) > 0$

- We saw an example where T-ZIP does poorly
- Even ZI does better!
- T-ZIP converges even when there is no equilibrium price

T-ZIP: $Var(pos) > 0$

- We saw an example where T-ZIP does poorly
- Even ZI does better!
- T-ZIP converges even when there is no equilibrium price
- This causes failure: harmful convergence

T-ZIP: $Var(pos) > 0$

- We saw an example where T-ZIP does poorly
- Even ZI does better!
- T-ZIP converges even when there is no equilibrium price
- This causes failure: harmful convergence
- Mechanics are quite complex, see thesis

Contents

Conclusions

Conclusions

- T-CDA allows efficient resource allocation, even with

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting
  - (CDA) Multiple users, multiple providers

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting
  - (CDA) Multiple users, multiple providers
  - (CDA) Users, providers enter/leave at any time

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting
  - (CDA) Multiple users, multiple providers
  - (CDA) Users, providers enter/leave at any time
  - (New) Execution Uncertainty

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting
  - (CDA) Multiple users, multiple providers
  - (CDA) Users, providers enter/leave at any time
  - (New) Execution Uncertainty
- Like CDA, T-CDA can be fully decentralised (peer-to-peer implementation)

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting
  - (CDA) Multiple users, multiple providers
  - (CDA) Users, providers enter/leave at any time
  - (New) Execution Uncertainty
- Like CDA, T-CDA can be fully decentralised (peer-to-peer implementation)
- Hence, solves stated problem

Conclusions

- T-CDA allows efficient resource allocation, even with
  - (CDA) Large problem space
  - (CDA) Non-cooperative setting
  - (CDA) Multiple users, multiple providers
  - (CDA) Users, providers enter/leave at any time
  - (New) Execution Uncertainty
- Like CDA, T-CDA can be fully decentralised (peer-to-peer implementation)
- Hence, solves stated problem
- No generally successful strategy

Discussion

Discussion

- Need a general strategy

Discussion

- Need a general strategy
    - Essential for complete evaluation of T-CDA

Discussion

- Need a general strategy
  - Essential for complete evaluation of T-CDA
  - This need was identified too late

Discussion

- Need a general strategy
  - Essential for complete evaluation of T-CDA
  - This need was identified too late
- Apply to more realistic settings

Discussion

- Need a general strategy
  - Essential for complete evaluation of T-CDA
  - This need was identified too late
- Apply to more realistic settings
- Develop a model that *learns* trust from market interactions